

Robust Feature Extraction for On-board Monocular-based Spacecraft Pose Acquisition

Vincenzo Capuano*

California Institute of Technology, Pasadena, CA, 91125, USA

Shahrouz Ryan Alimo†

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109, USA

Andrew Q. Ho‡

California Institute of Technology, Pasadena, CA, 91125, USA

Soon-Jo Chung§

California Institute of Technology, Pasadena, CA, 91125, USA

This paper presents the design, implementation, and validation of a robust feature extraction architecture for real-time on-board monocular vision-based pose initialization of a target spacecraft in application to on-orbit servicing and formation flying. The proposed computer vision algorithm is designed to detect the most significant features of an uncooperative target spacecraft in a sequence of two-dimensional input images that are collected on board the chaser spacecraft. A novel approach based on the fusion of multiple and parallel processing streams is proposed to filter a minimum number of extracted true point features, even in case of unfavourable illumination conditions and in presence of Earth in the background. These are then combined into relevant polyline structures that characterize the true geometrical shape of the target spacecraft.

I. Introduction

Estimating the pose (relative position and attitude) of a target spacecraft or neighboring spacecraft is an important problem for a wide range of mission scenarios, such as formation flying (FF) [1, 2], on-orbit servicing (OOS) [3], active debris removal (ADR) [4]. Relative pose estimation and navigation is indispensable to safe close-proximity operations (from a few meters to several tens of meters), such as rendezvous, final approach and docking, station-keeping, and monitoring. In particular, relative navigation maneuvers must be performed autonomously without the use of ground control stations. Indeed, relative navigation maneuvers based on ground commands would be infeasible or difficult due to unavoidable communication delays or the lack of satellite coverage and in situ information.

Several approaches and metrology systems can be used for spacecraft pose determination, depending on whether the target is actively cooperative, passively cooperative, uncooperative, known or unknown [5]. In the most general case of uncooperative target (i.e., it is not equipped with any artificial marker and any communication link), a vision-based approach can be adopted for its pose estimation. Unlike LIDARs, Vision-based pose estimation involves the use of lower power passive electro-optical sensors (monocular or binocular cameras), also compatible with small and very small platforms (e.g., nanosatellites, picosatellites, and femtosatellites) [1, 6, 7].

In this paper, we focus on monocular vision-based pose determination, since a single camera with vision processing offers a low-cost solution that works for close to far-distance ranges. As illustrated in Fig. 1, the goal of pose determination is to estimate the translation \vec{T}_{BC} and rotation R_{BC} between the body frame of the target and the camera frame on board the spacecraft chaser, by processing the pixels in the image frame. The vision-based pose determination process consists of three main steps (see Fig. 2): i) image processing (and feature extraction), ii) feature matching, and iii) estimation of the transformation between the matched features. While in our previous research [5] we mainly investigated the last two

*Postdoctoral Researcher, Graduate Aerospace Laboratories of California Institute of Technology (GALCIT), vcapuano@caltech.edu

†Lead Machine Learning Scientist, Deep Learning Technology Group at Jet Propulsion Laboratory (JPL), Shahrouz.R.Alimo@jpl.nasa.gov

‡Undergrad Student, Computer Science, California Institute of Technology, aqho@caltech.edu

§Associate Professor of Aerospace and Bren Scholar and JPL Research Scientist, Graduate Aerospace Laboratories of California Institute of Technology (GALCIT), sjchung@caltech.edu

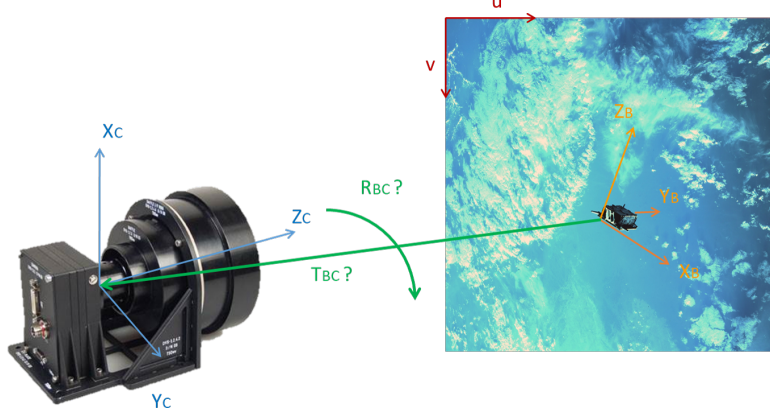


Fig. 1 Camera Frame, Image Frame, Body Frame and Pose Estimation Unknowns.

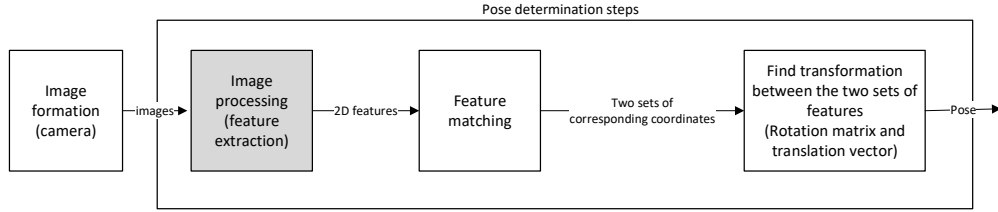


Fig. 2 Vision-based Pose Estimation Steps.

steps, in this paper, we focus on the problem of developing a robust image processing strategy required to extract a number of natural features of the target. In particular, the attention is placed on the initialization phase of the pose estimation process, when no a-priori knowledge of the pose is available.

Processing effectively actual 2D images of a spacecraft for pose initialization can be extremely challenging since (as shown in Fig. 4) these can be characterized by low signal-to-noise ratios and high image contrasts due to poor and extremely variable light conditions and by the presence of Earth (or other celestial bodies or objects) in the background, which can give false or partial feature detection, resulting in the extraction of an insufficient number of features, required for its pose acquisition. In addition, the adopted algorithms must be compatible with limited on-board computational resources, in order to be able to run on board, in real time.

We propose a robust feature extraction architecture for real-time on-board monocular vision-based pose initialization of an uncooperative target spacecraft. The proposed vision processing strategy can detect the most significant features in the two-dimensional input image, by first fusing multiple and parallel feature extraction streams to identify a minimum number of extracted true point features, even in case of unfavourable illumination conditions and in presence of Earth in the background. The filtered point features are then combined into line segments and more complex geometrical structures, composition of line segments or polylines, such as polygons that characterize the true geometrical shape of the target spacecraft (e.g., end points of body edges, solar panels, antennas). The proposed feature extraction approach is particularly advantageous when adopting a model-based pose determination technique. Indeed, as suggested and highlighted in [8, 9], the computational burden of the image-to-model matching process can be greatly reduced by searching for the correspondence between a smaller number of more complex geometrical structure, i.e. polylines, polygons, rather than between a higher number of extracted point features. In particular, a simplified model can be built before the mission, which only includes the point features and polylines features that most likely will be detected, given the geometry of the target, the resolution of the images, the approximate distance of the target and the operational lighting condition.

The proposed strategy is validated by processing actual space images collected during the *PRISMA Formation Flying* mission [10], which was a technology demonstration mission for the in-flight validation of sensor technologies

and guidance/navigation strategies for spacecraft formation flying and rendezvous. The images of the target spacecraft called *Tango*, processed in the present paper, were collected by the *Techno System Development* camera system [11] on board the main spacecraft called *Mango*, during proximity maneuvers in far, mid, and close ranges.

The paper is organized as follows. Following this introduction, in Section I a short overview of the related works and the problem statement are provided. Section II describes the high level architecture of the proposed approach. Then, the adopted algorithms are outlined in more details in Section III. Section IV presents a preliminary analysis of the performance of the proposed strategy. Finally in Section V, we draw the conclusions.

A. Related work

Recent progress in computer vision and image analysis have enabled the implementation of advanced monocular-based relative navigation algorithms. With the expanding use of self-driving cars, this field of research is growing very fast, offering many solutions to the problem of robust image feature detection and segmentation.

A few prior studies have been carried out so far that specifically focus on feature extraction for vision-based spacecraft pose estimation. In particular, a monocular based spacecraft pose initialization approach, based on Harris corner detector [12] was proposed in [13]. [14] blended two common feature detectors, namely the Shi-Tomasi corner detector [15] and Speeded Up Robust Features (SURF), which is a rotation and scale invariant detector [16], to improve the robustness of the edge detection in different illumination conditions. To resolve this issue, a background subtraction and Gaussian blob detection algorithms were adopted to provide relative distance information for low-resolution camera images. In the interesting work [8], Sobel and Hough transform algorithms [17] were used to detect perceptual groups of the target's true edges. The approach is refined in the most recent study [9], with a more complex image processing architecture based on an image gradient-based filter, which is used to distinguish Earth in the background of the target.

Our approach makes use of an image gradient-based filter similar to the “Weak Gradient Elimination” proposed in [9] for foreground-background segmentation and for region of interest (ROI) detection. However, in contrast with the prior works, in our approach the extracted ROI is processed adopting three feature extraction streams, whose outputs are then filtered in point features and combined into polyline structures. In addition, in contrast to [9], our architecture does not rely on the Sobel operator and makes use of a different strategy to merge multiple detected segments.

B. Problem Statement

Definition 1. Let x_0, x_1, \dots, x_N be a sequence of points in a plane. A polyline ^{*} is defined as the union of the line segments $x_0x_1, x_1x_2, \dots, x_{N-1}x_N$ (see Fig. 3). The points x_i are named as vertices of the polyline and the line-segments $x_i x_{i+1}$ are named as edges of the polyline.

According to Definition 1, we consider the projection of the target spacecraft onto the image frame, as a structure that consists of geometrical shapes such as *polylines*, as well as other possible curvilinear shapes, such as circumferences, parabolas, etc. In this work, we consider images of the target spacecraft acquired in unfavorable illumination and background conditions. However, we also assume that the number of pixels in the region of interest (ROI) is big enough to allow the extraction of a minimum number of point features required for pose determination of the target spacecraft. The problem statement can be stated as: given a two-dimensional image of the target spacecraft, the goal is to extract a number of true points that belong to the target geometrical structure and then to identify among them the ones part of its *polylines*.

II. Architecture Overview

In this section, we present an overview of our approach. Figure 5 illustrates the high-level architecture designed to perform feature extraction from actual images of an orbiting artificial satellite. An adaptive strategy was conceived to provide a feature extraction capability, robust and flexible, effective in case of low signal-to-noise ratios, due to poor illumination conditions, and at the same time also in presence of Earth (or any other known planetary body) in the background of the target.

As shown in Fig. 5, in a first step the raw image is preprocessed. The preprocessing includes the lens distortion correction that takes into account the camera parameters, a Gaussian filter used to reduce the noise in the rectified image, and therefore its conversion to gray scale. Then, one of two alternative approaches is selected if Earth is predicted to be

^{*} The polyline is also known as polygone curve [18]. Also, polyline concept is defined in the computer graphics and commercial software like AutoCAD.

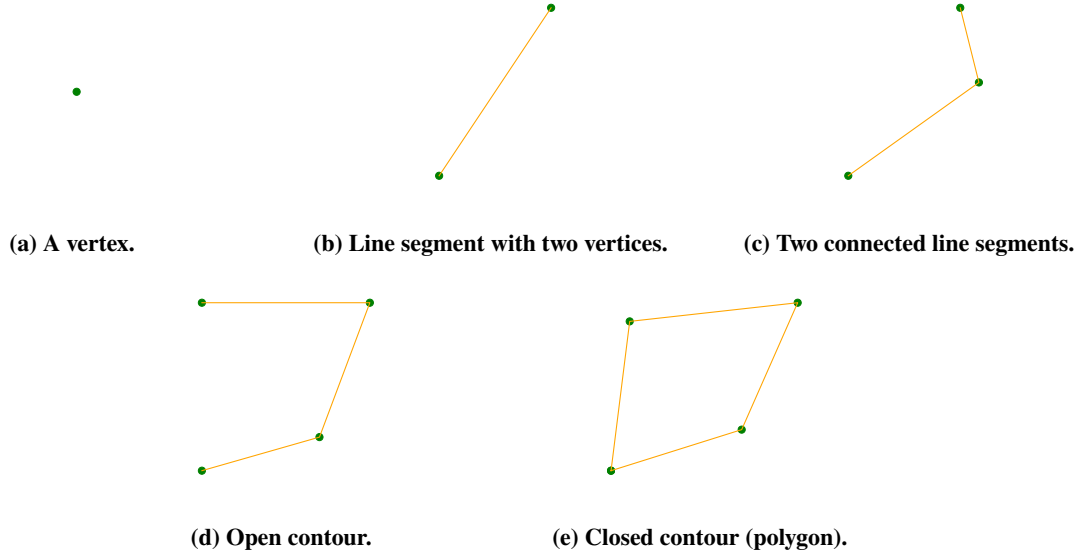


Fig. 3 Different types of polyline.

present or absent in the background. Respectively, one approach is based on the image gradient and the other one on the image thresholding. Hence, both approaches rely on a combination of image processing algorithms organized in multiple processing streams, whose extracted feature outputs are therefore fused together. The goal of the fusion is to minimize the outliers and reduce the inliers to only the most significant ones, among the extracted features.

A. Feature Extraction in Presence of Earth in the Background

In presence of Earth in the background (see Fig. 4a, 4b, and 4c), preventing false detection and extracting true features only of the target spacecraft can be very challenging or nearly impossible. Additional features of the planet's surface in the background might be detected because of their morphologic and photometric characteristics. For example, line segments belonging to clouds, coastlines or horizon might be mistakenly extracted using a segment detector while many true features of the target might be not extracted because of the higher reflectivity of clouds and ocean.

Figure 6 shows the gradient-based strategy adopted in presence of Earth in the background. To do so, we calculate the image gradient as proposed in [9] and then filter out all pixels for which the gradient magnitude is lower than a certain threshold. The image output is therefore the input of three different parallel processing streams:

- 1) In a first stream we detect the ROI, then used to crop the preprocessed image on which we extract a number (empirically determined) of strongest Shi-Tommasi corners [15];
- 2) In the second stream we extract segments and their end points by using the Hough transform [20];
- 3) In the third stream we extract segments and their end points by using Line Segment Detector (LSD) [13].

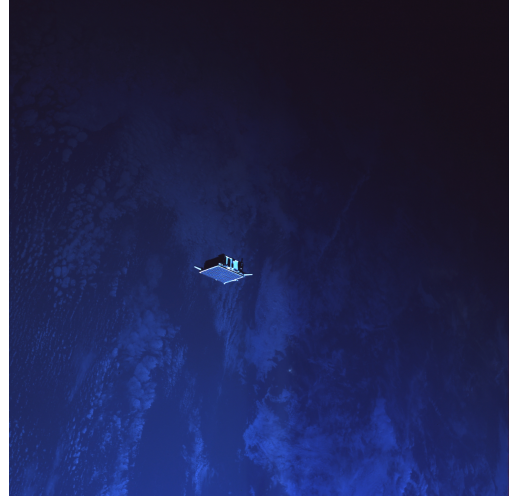
The points detected in the three streams are filtered and then combined into more complex polylines, as described in greater detail, in Section III. In this way, points are combined into segments, which are merged into chains of consecutive segments (polylines), which if possible are combined to form polygons (closed polylines).

B. Feature Extraction in Absence of Earth in the Background

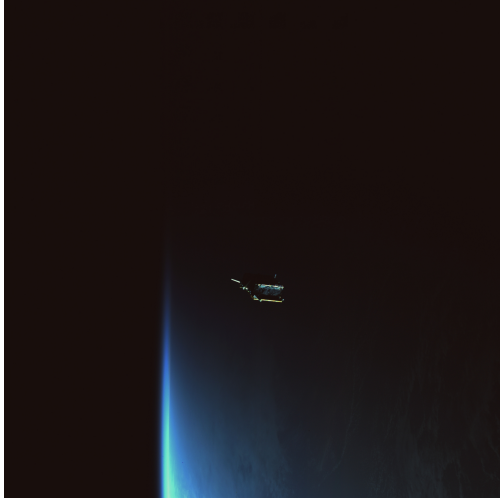
In absence of Earth in the background, we preprocess the raw image and then use a thresholding method, which results to be more efficient than the gradient-based filter when the photometric characteristics of the background can be easily distinguished from the ones of the foreground, as well as less computationally expensive as highlighted in Section IV. The output is then processed to find the ROI. The portion of image in the ROI is therefore the input of the same three processing streams, point filtering and polylines synthesis algorithms adopted in presence of Earth in the background.



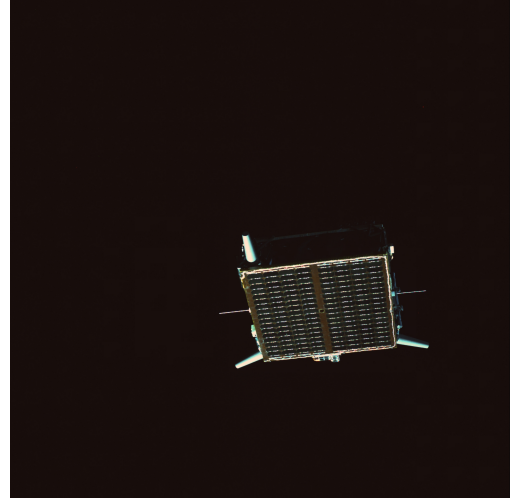
(a) Earth in the background. Strong image contrasts.



(b) Earth in the background. Weak image contrasts.



(c) Earth horizon in the background.



(d) Absence of any visible object in the background.

Fig. 4 Different background illumination conditions in actual image of an artificial satellite, a Tango [19].

III. Algorithms Description

In this section, we describe the algorithms adopted in the proposed architecture.

A. Background Removal

We first load the image and apply the Prewitt Operator to find the gradient. Using the result of the Prewitt Operator, we use an image gradient-based filter to remove unnecessary features in the background. We calculate the image gradient using the Prewitt Operator proposed in [9] to filter out all pixels for which the gradient magnitude is lower than a certain threshold q . The threshold is calculated as the quantile of all the gradient magnitude values in the image for the cumulative probability or probabilities 0.999 in the interval $[0,1]$. A value of 0.99 is instead used to provide the input of the three parallel processing streams.

B. Cluster-based Thresholding

In absence of Earth in the background, there is a more evident difference between the target spacecraft and its background in the gray scale image and it is possible to classify each pixel in the image respectively as foreground or background. In this regard, we use a histogram Gaussian Mixture Model (GMM) method [21], a histogram based

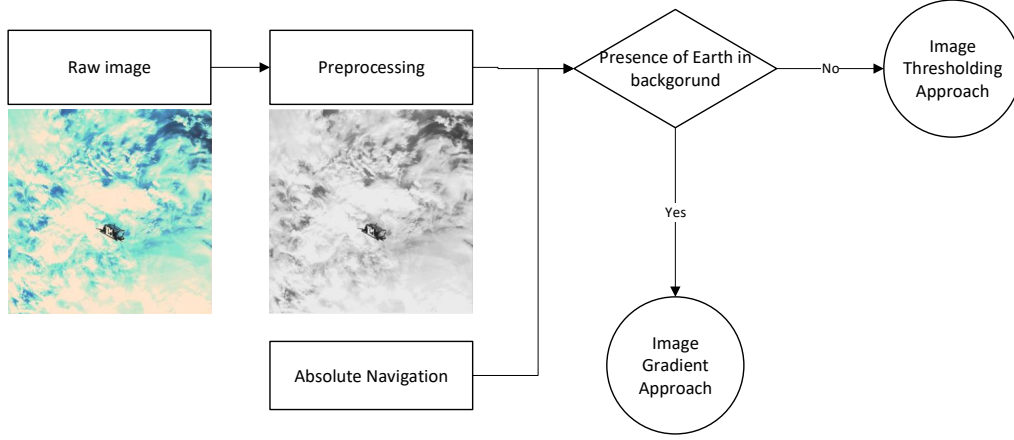


Fig. 5 High-level approach.

method for clustering the data into different classes. The Expected Maximization (EM) is used to label the image pixels, X , into two classes of background and foreground, so that their intra-class variance is minimized. Denoting the posterior distribution as $P(c|x)$, where c is the label of x , and considering that $c = 1$ for the foreground and $c = 0$ for the background, we determine if a pixel, y , is either part of the background or not by using the Bayesian solution as follows:

$$\arg \max_{i \in \{0,1\}} \frac{P(x|c=i)P(c=i)}{P(x)}. \quad (1)$$

In (1), the posterior distribution is assumed to be Gaussian, $P(x|c=i) = \mathcal{N}_i$, and the prior to be $P(c=i) = \pi_i$ in which $i \in \{0,1\}$. In this situation, the histogram of the image pixel intensity can be represented as:

$$P(X) = \pi_0 \mathcal{N}_0 + \pi_1 \mathcal{N}_1. \quad (2)$$

In order to find the optimal weights π_i to cluster the image into foreground and background, we use the Otsu's clustering thresholding scheme, which minimizes the weighted sum of within-class variances of the foreground and background pixels to establish an optimum threshold for clustering. The Otsu's clustering thresholding scheme is discussed in detail in [22].

C. Region of Interest Segmentation

We calculate the ROI as the smallest zone that bounds the remaining pixels from gradient-based filter or from Otsu's thresholding. We iterate through the pixels of the image and store where the top-most, bottom-most, left-most, and right-most non-zero pixels in the image occur, thereby enclosing the smallest rectangular region that contains the target spacecraft. The resulting rectangle ROI is used throughout the rest of the algorithm.

D. Parallel Stream Processing

The heart of our approach lies in point filtering. Using the three sets of points produced by Hough Line Transform (HLT), Line Segment Detector (LSD), and Shi-Tomasi corner detection, only points that are common to all three sets are retained for feature synthesis.

1. Hough Line Transform

Hough Line Transform (HLT) takes a binary image as input and returns a list of detected line segments. HLT is one of the most popular line detection techniques that is used for feature extraction for artificial satellites, providing accurate results for a wide variety of images [8, 9]. One of the main challenges for Hough transform is tuning hyperparameters (e.g., distance resolution, angle resolution, an accumulator threshold parameter, minimum line length, and maximum line gap) in an efficient and accurate way for each image. This fine tuning can be easily be automated using blackbox

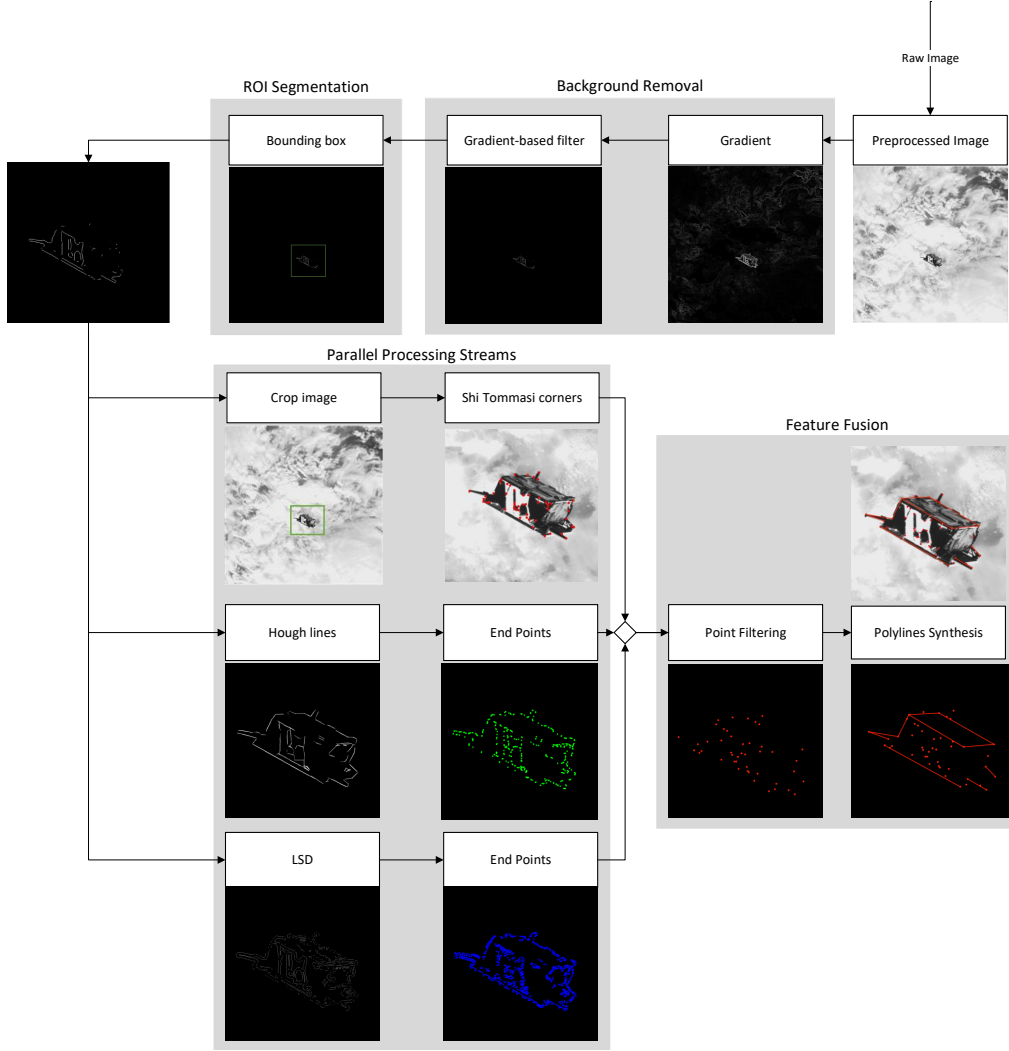


Fig. 6 High level architecture in presence of Earth in the background.

optimization schemes such as [23, 24]. One of the most common variants of the HLT is the Probabilistic HLT [20], in which a random, smaller subset of the original significant points is used for computation instead. Although the probabilistic Hough transform seems to have better performance than the standard Hough transform [20], it still requires the tuning of hyperparameters.

We use Probabilistic HLT as one of the approaches to identify polylines associated with the spacecraft and to increase efficiency at the minimal expense of accuracy for a binary image. One of the disadvantages of the Probabilistic HLT is the non-deterministic results of the output, which means that different runs may yield slightly different results due to the stochastic nature of this approach.

The random sampling approach used by Probabilistic HLT in addition to its reliance on hyperparameter tuning introduces the issue of detecting lines that do not exist. For many images, an incorrect subset of points may accidentally satisfy the collinearity condition of Hough Transform for a specific set of parameters. Thus, while HLT returns significant, useful edges, we must also consider the fact that erroneous edges are also included in the results.

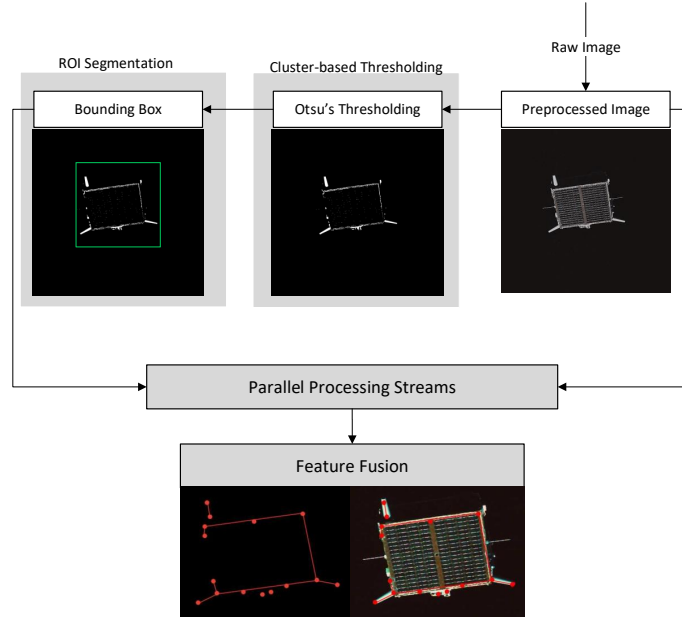


Fig. 7 High Level Architecture in Absence of Earth in the Background.

2. Line Segment Detector

Line Segment Detector (LSD) [13] takes a gray-scale image as input and returns a list of detected line segments. It is designed to detect locally straight contours that represent the line segments by searching in different rectangular boxes. Unlike HLT, LSD does not require hand-tuning the hyperparameters, and returns subpixel accurate results. However, the set of line segments generated by LSD often includes edges that do not contribute to the desired significant features on the spacecraft.

3. Shi-Tomasi Corner Detection

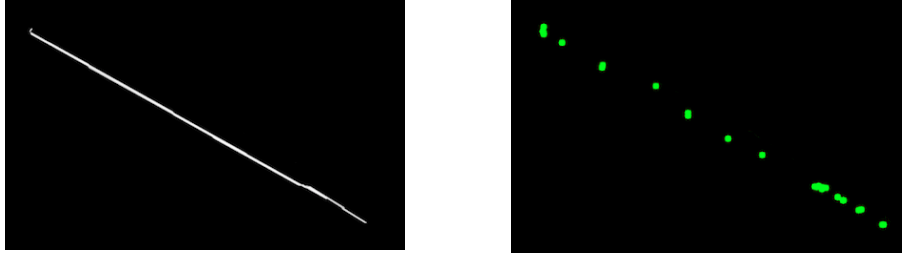
Shi-Tomasi (ST) [15] detects a set of corner points in an image. This method is nearly identical to the Harris corner detector, differing only in its scoring criteria. The score function in this scheme is defined as minimum of the eigenvalue of the gradient of the image. This approach, used to detect meaningful corners of an object, is widely used in feature tracking. However, ST corner detector suffers from inaccuracy with regards to sharp speckle noise. Due to the nature of the algorithm, sharp, random noise often triggers a false positive, as these small random spots may have large eigenvalues from gradient calculation.

E. Feature Fusion

The points detected in the three streams are filtered and then combined into more complex geometrical structures by exploiting geometrical conditions, as mentioned in Section II. Although HLT detects segments that do not exist as a result of randomization, LSD detects segments that are unnecessary for polyline construction, and Shi-Tomasi often finds corners from random sharp specks that are undetected by HLT and LSD, taking the intersection between HLT points, LSD points, and ST points effectively neutralizes a majority of the noise contributed by each set. Erroneous edges from HLT are not present in detected edges in LSD, and insignificant edges from LSD are not present in HLT. In addition, specks with strong intensity that ST is susceptible to will not appear in HLT nor LSD edges, and any potential incorrect agreement between pseudo-collinear points from HLT and LSD is ruled out by ST.

Previous approaches to the pose estimation problem have relied on singular techniques such as HLT and Corner Detection algorithms for example. However, as already highlighted, each of these approaches has different detriments. In particular, HLT possesses several drawbacks, such as the necessity of fine-tuning parameters on a case-by-case basis, making HLT somewhat challenging to work with for auto-detection of features. The nature of the algorithm makes it

particularly susceptible to jagged edges, as this increases the number of potential edge points, breaking up significant long segments into smaller segments, and opening the possibility of overlapping edges. For example, consider the following jagged edge taken from a real data-set, and the corresponding Hough Line detected in Fig. 8a and 8b.



(a) Overlapping detected Hough line Segments (b) Endpoints of Detected Hough Line Segments

Fig. 8 Jagged and overlapping segments detected with HLT.

Although the edge appears mostly straight, close inspection reveals that it is actually composed of several overlapping lines. While it is possible for an algorithm to iterate through all detected segments of Fig. 8a and 8b and merge them via adjacency and angular geometric conditions, it is much more difficult to do so accurately in the case of disjoint lines, such as the ones detected in Fig. 9a. For this set of edges, it is nearly impossible to reliably merge the segments detected using HLT, by only exploiting geometric conditions, such as reasonable adjacency, without compromising accuracy or fine tuning of HLT parameters. However, while the edges themselves may not be adjacent, the points are still collinear, as shown in Fig. 9b. We aim to exploit the relative geometry of HLT and LSD end points to build significant polylines.

Consider the overlaid set of points from HLT (Green), LSD (Blue), and ST (Red) in Fig. 11a. Overlapping points from all three methods are circled in white. To find the intersection between all three sets of points, we implement a K-D spatial partitioning tree and a K-Nearest neighbor search. As we can control the number of ST points we extract, we construct K-D trees from HLT and LSD points and match ST points to these trees. For example, consider the HLT points shown in green in Fig. 11b. We divide the points by alternating axes, first splitting on the y-axis, and partition each resulting zone by the other axis to separate points into a spatially-oriented binary tree. Through this representation of points, we can easily apply a K-Nearest Neighbor (KNN) search with $O(\log(n))$ run time, scaling with number of points, to find the intersection of points and form the “Filtered Point Set.”

The last step is to extract edges that should exist within the image by connecting sets of filtered points, using the output of gradient-based filter or image thresholding as a scaffold. Pairs of filtered points are evaluated for edge synthesis candidacy by representing the line passing through in parametric form with a single step difference the distance between the two points, as shown in Fig. 11e. The line segment is divided into test points and the intensity value of the threshold output at that specific pixel for each is measured. If the number of non-zero pixels exceeds that of the threshold, then an edge is detected.

Individual detected edges can be conjoined to one single edge. For example, consider Fig. 11f, in which an edge between P_1 and P_2 has already been stored, and an edge between P_1 and P_3 is being tested for existence. We test for collinearity by calculating the normal distance of P_1 from the line segment P_2 and P_3 as follows:

$$D_{P_1} = \frac{|(y_3 - y_2)x_1 - (x_3 - x_2)y_1 + x_3y_2 - y_3x_2|}{\sqrt{(y_3 - y_2)^3 + (x_3 - x_2)^2}}. \quad (3)$$

If the distance relative to the length of the line segment is small enough, then we conjoin P_2 and P_3 into a single edge, removing the middle point P_1 , as demonstrated in Fig. 11g. Conversely, we also may run into the issue of overlapping edges. In Fig. 11h, the tested edge in dashed green from P_1 to P_3 overlaps with the solid blue previously detected edge from P_1 to P_2 . If the angle between the two segments is small enough, then we have found two overlapping edges originating from the same point, P_1 . We then retain the longer, significant edge, as shown in Fig. 11i.

IV. Performance

This section presents preliminary results obtained when using the proposed feature extraction strategy to process actual images of the target spacecraft *Tango*, collected by the camera [11], on board the main spacecraft *Mango* during the mission *PRISMA*. The camera [11] has a focal length of 30 mm and based on a colour camera with a CCD sensor,

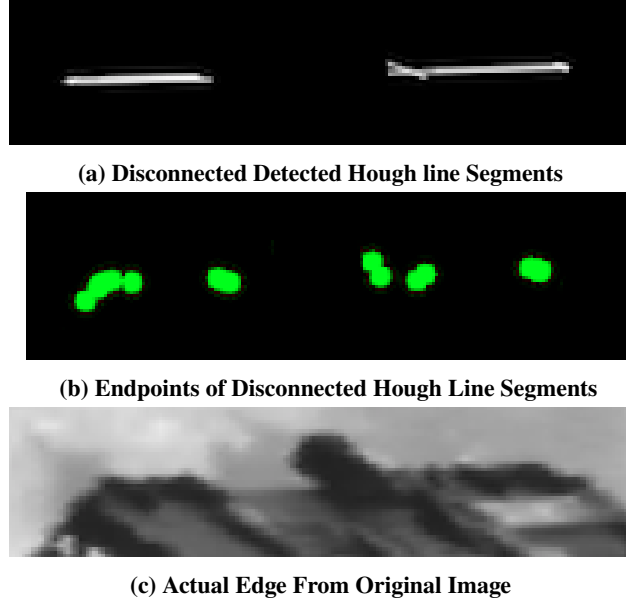


Fig. 9 Comparison of detected edges and endpoints in relation to the true edge of the spacecraft

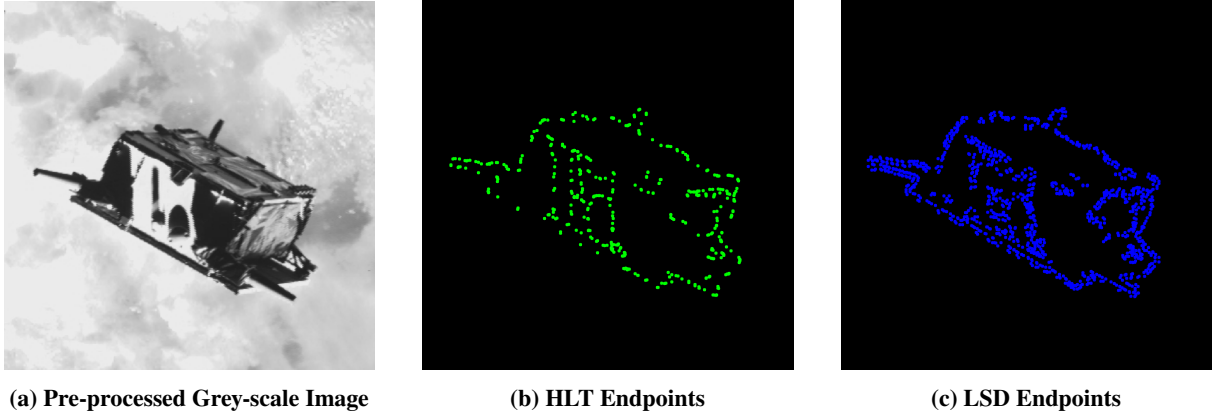


Fig. 10 Comparison of detected edges and endpoints in relation to the true edge of the spacecraft.

acquires colour images and video sequences employing a Bayer (RGB) filter, with a resolution of 2048×2048 pixels, each of $7.4 \times 7.4 \mu\text{m}$. The target spacecraft Tango is very small and its geometry can be simplified in a convex polyhedron representing the spacecraft body of $560 \times 550 \times 300$ mm, a polygon representing the solar panel of 560×750 mm and five segments each representing a line antenna of 204 mm [9].

A. Preliminary Results

Figure 12 displays the obtained results on four selected images, each one part of a different considered sequence (of about 30 images), representative of different possible lighting and background conditions and of different camera/target separations between 5 and 45 m. In our our analysis, we measured the accuracy of the feature extraction by calculating the euclidean distance between the detected point feature and its corresponding ground truth. A point feature is considered to be a true positive if detected with a minimum accuracy of 7 pixels (which corresponds to different values in cm depending on the camera/target separation).

Figure 12b shows the features extracted in the image of Fig. 12a, in absence of Earth in the background, for a camera/target separation of about 9 m. All the detected point features are part of the target and are extracted with a maximum error of 1 cm approximately. In addition, the most significant point features are detected, as part of true

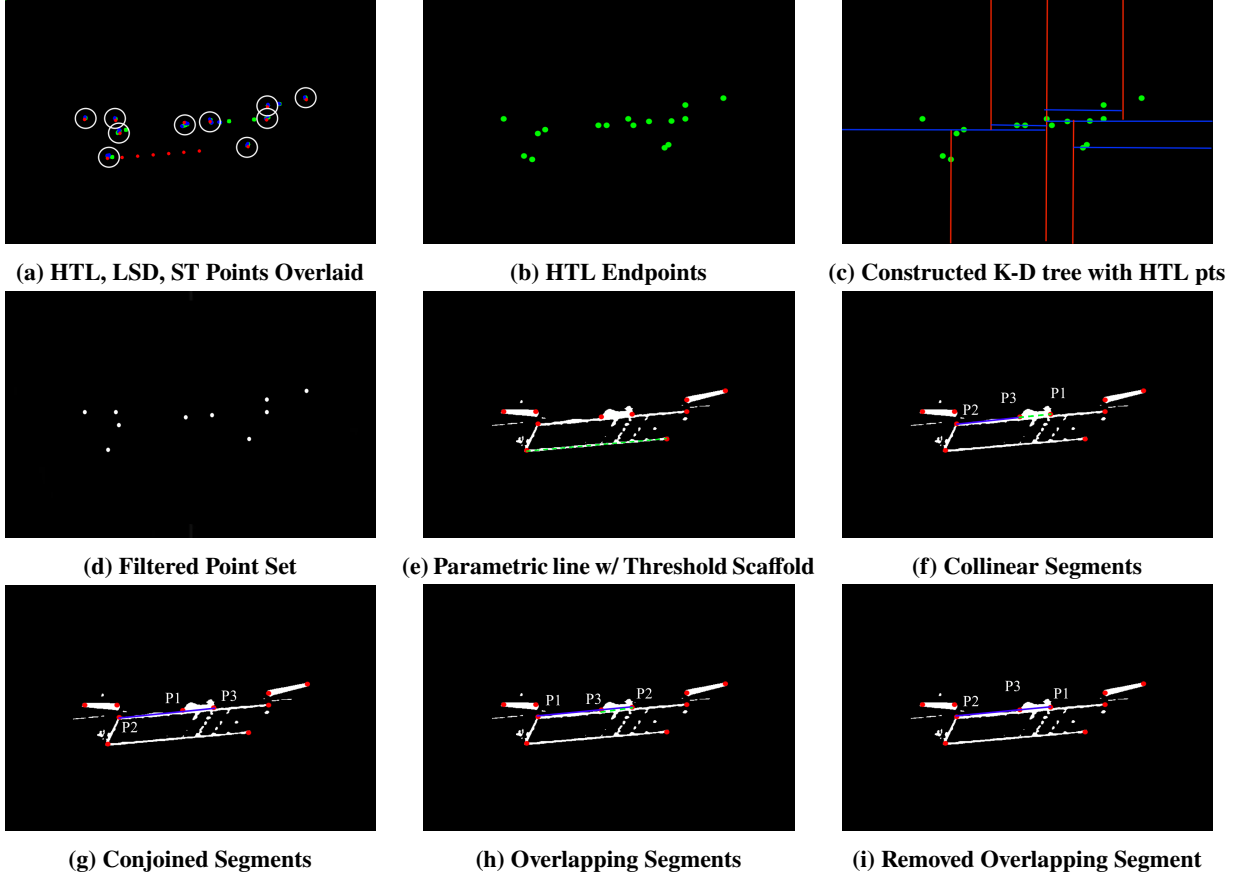


Fig. 11 Comparison of detected edges and endpoints in relation to the true edge of the spacecraft.

identifiable polylines of the target spacecraft. The image in Fig. 12c is instead characterized by the presence of Earth in the background, with similar photometric characteristics as the ones of the target spacecraft in the foreground. The target is at approximately 16 m from the camera. Due to the strong reflection of clouds and ocean, using a traditional feature extraction methods would result in a large number of false positives. As shown in Fig. 12d, thanks to a gradient-based filter, no false feature in the background is detected and in addition, thanks to the fusion of three feature extraction streams, only the most significant true features of the target spacecraft are detected with a maximum error of 3.5 cm. However, the image output also contains one spurious edge in the right bottom. Figure 12f illustrates the extracted features from the image of Fig. 12c, which is characterized by low signal-to-noise ratio and presence of Earth horizon in the background. In this image the target is 20 m far away from the camera. Despite the sharp change of image intensity in the horizon line, the use of a more selective threshold in the gradient-based filter allows for successful background removal. Because of the poor illumination conditions, about 50% of extracted filtered point features are false positive; however 100% of the extracted point features that belong to the identified polylines are true positive, with an accuracy of 3.9 cm. Finally, Figure 12h displays the features detected in Figure 12g, when the target is about 45 m far away from the camera. Only 9 point features are detected in total, 8 are part of identified polylines. The maximum detection error is 2.2 cm.

By considering all processed images, approximately only 7% of the detected polylines and less than 5% of the detected polyline vertices are false positive.

B. Computational Time

Figure 13 illustrates the relative run time for each processing step adopted in presence of Earth in the background. The gradient-based ROI segmentation (gradient-based background removal and ROI segmentation) results to be the most computationally expensive step, taking 85% of total time complexity. The average run time of this step is 0.3 s

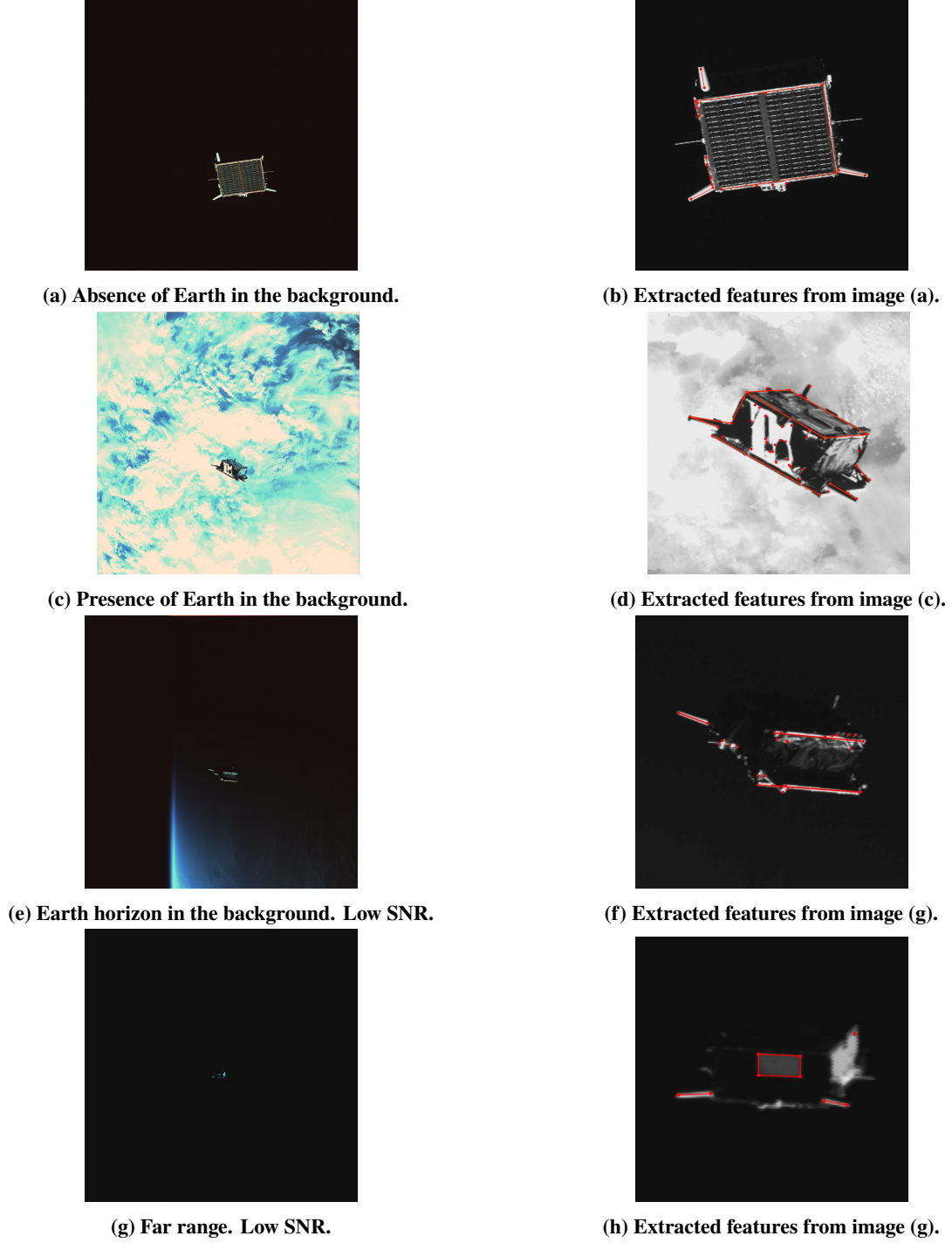


Fig. 12 Different background illumination conditions in actual image of an artificial satellite, a Tango representative of different background and illumination conditions [19].

when implemented in MATLAB on a 1.8 GHz Intel Core i7-8550U processor for a 2048×2048 pixels RGB image. When using lower resolution images, as expected the run time is significantly smaller (e.g. 0.07 s for a 752×580 pixels image). As also highlighted in [9], the image-gradient based filter reduces the number of pixels to be further processed. Therefore using three streams to process a smaller number of pixels does not affect significantly the computational burden. For example, we measured the run time of other state-of-the art feature extraction algorithms, as Canny edge

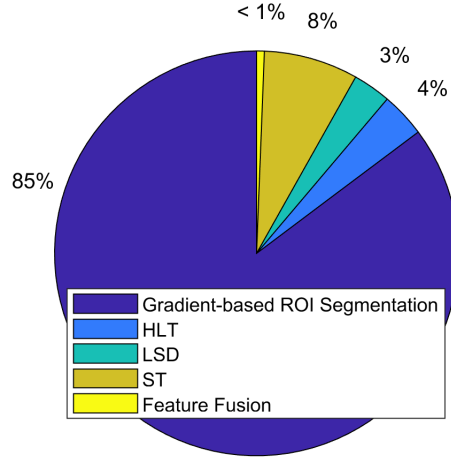


Fig. 13 Relative run time of each processing step of Figure 6.

detector, BRISK, FAST, SURF feature detectors and Harris and Shi-Tommasi corner detectors, when used to process the whole 2048×2048 image of Fig. 12c. Except for the FAST feature detector that is about one order of magnitude faster, the computational time of the proposed architecture (which uses multiple processing streams on the smaller ROI) is about 15% lower than the average of the other feature detectors. Similarly, in absence of Earth in the background, the run time of cluster-based thresholding and ROI segmentation is 75% of the total run time, however, 60% lower than the gradient-based ROI segmentation.

V. Conclusion and Future Work

This paper presented a robust feature extraction architecture developed for real-time on-board monocular vision-based pose initialization of target spacecraft, which could be used for proximity navigation for on-orbit servicing and formation flying. The proposed approach is based on the fusion of multiple and parallel vision processing streams to select a minimum number of extracted true point features, even in case of unfavourable illumination conditions and in presence of Earth in the background. These are combined into relevant polyline structures that characterize the true geometrical shape of the target spacecraft. Identifying true polylines of the target geometry allows for a significant reduction of the search space in the image-to-model matching in model-based pose determination approaches, with a consequent saving in computational burden. The benefit of the proposed strategy can be maximized if a simplified geometric model is built taking into account the polylines that most likely will be detected. In addition, note that the proposed strategy can also be adopted in case of unknown target, in a SLAM-based approach [5]. Indeed, tracking a smaller number of polyline features instead of a larger number of point features can reduce the computational load.

An overview of the high level architecture and a more detailed description of the adopted algorithms were provided in this paper. Preliminary results were presented, obtained processing actual space images of the small satellite Tango, collected during the mission PRISMA. The proposed strategy seems to be promising; with a maximum error (euclidean distance from the corresponding ground truth) of 7 pixels, only 7% of the detected polylines and less than 5% of the detected polyline vertices are false positive. However, depending on the distance of the target, the 7 pixels error in the image frame is equivalent to an error of several centimeters in the 3D camera frame, with a consequent impact on the final achievable pose determination accuracy. Anyhow, once the pose determination process has been initialized, it is possible to refine the estimation in tracking, exploiting the apriori knowledge of the target pose.

The total computational time of our approach scales most heavily by the total number of pixels in the image and it results to be nearly equivalent to the one of most of the state-of-the-art methods adopted for real-time pose estimation.

For future work, we plan to validate the proposed architecture by processing a dataset of both actual and synthetic images to rigorously and quantitatively evaluate the performance in terms of accuracy, robustness and sensitivity of the feature extraction. A fourth processing stream could be added to detect cylindrical, spherical, and circular features, often also identifiable in the geometry of an artificial satellite. Moreover, we will focus on feature tracking as well as on feature extraction, making use of a model of the chaser/target relative dynamics.

Acknowledgments

The first author would like to thank the Swiss National Science Foundation (SNSF), that supported him for this research. The authors also gratefully acknowledge partial funding from Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (NASA) in support of this work. The authors thank F. Y. Hadaegh, A. Stoica, and M. Wolf. In addition, the authors would like to thank Techno System Development s.r.l. that kindly made the images of Tango available for this research study.

References

- [1] Hadaegh, F. Y., Chung, S.-J., and Manohara, H. M., "On development of 100-gram-class spacecraft for swarm applications," *IEEE Systems Journal*, Vol. 10, No. 2, 2016, pp. 673–684.
- [2] Chung, S.-J., Ahsun, U., and Slotine, J.-J. E., "Application of synchronization to formation flying spacecraft: Lagrangian approach," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, 2009, pp. 512–526.
- [3] Flores-Abad, A., Ma, O., Pham, K., and Ulrich, S., "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, Vol. 68, 2014, pp. 1–26.
- [4] Liou, J.-C., and Johnson, N. L., "A sensitivity study of the effectiveness of active debris removal in LEO," *Acta Astronautica*, Vol. 64, No. 2-3, 2009, pp. 236–243.
- [5] Capuano, V., Kim, K., Hu, J., Harvard, A., and Chung, S.-J., "Monocular-Based Pose Determination of Uncooperative Known and Unknown Space Objects," 2018.
- [6] Atanasov, N., Sankaran, B., Le Ny, J., Pappas, G. J., and Daniilidis, K., "Nonmyopic view planning for active object classification and pose estimation," *IEEE Transactions on Robotics*, Vol. 30, No. 5, 2014, pp. 1078–1090.
- [7] Bandyopadhyay, S., Foust, R., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Review of formation flying and constellation missions using nanosatellites," *Journal of Spacecraft and Rockets*, , No. 0, 2016, pp. 567–578.
- [8] D'Amico, S., Benn, M., and Jørgensen, J. L., "Pose estimation of an uncooperative spacecraft from actual space imagery," *5th International Conference on Spacecraft Formation Flying Missions and Technologies*, 2013.
- [9] Sharma, S., Ventura, J., and D'Amico, S., "Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous," *Journal of Spacecraft and Rockets*, 2018, pp. 1–16.
- [10] "PRISMA (Prototype Research Instruments and Space Mission technology Advancement)," <https://directory.eoportal.org/web/eoportal/satellite-missions/p/prisma-prototype>, 2018. Online; accessed 29 November 2018.
- [11] Capuano, G., Severi, M., Cacace, F., Lirato, R., Longobardi, P., Pollio, G., DeNino, M., and Ippolito, M., "Video system for prisma formation flying mission," *Proceedings of the IAA symposium on small satellite systems and services (4S)*, Rhodes, Greece, 2008.
- [12] Harris, C., and Stephens, M., "A combined corner and edge detector," *Alvey vision conference*, Vol. 15, Citeseer, 1988, pp. 10–5244.
- [13] Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G., "LSD: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 32, No. 4, 2010, pp. 722–732.
- [14] Cho, D.-M., Tsiotras, P., Zhang, G., and Holzinger, M., "Robust feature detection, acquisition and tracking for relative navigation in space with a known target," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 5197.
- [15] Shi, J., and Tomasi, C., "Good features to track," Tech. rep., Cornell University, 1993.
- [16] Bay, H., Tuytelaars, T., and Van Gool, L., "Surf: Speeded up robust features," *European conference on computer vision*, Springer, 2006, pp. 404–417.
- [17] Szeliski, R., *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.
- [18] Gomes, J., Velho, L., and Sousa, M. C., *Computer graphics: theory and practice*, CRC Press, 2012.
- [19] Gill, E., Montenbruck, O., and D'Amico, S., "Autonomous formation flying for the PRISMA mission," *Journal of Spacecraft and Rockets*, Vol. 44, No. 3, 2007, pp. 671–681.

- [20] Matas, J., Galambos, C., and Kittler, J., “Robust detection of lines using the progressive probabilistic hough transform,” *Computer Vision and Image Understanding*, Vol. 78, No. 1, 2000, pp. 119–137.
- [21] Alimohammadi, S., and He, D., “Multi-stage algorithm for uncertainty analysis of solar power forecasting,” *Power and Energy Society General Meeting (PESGM), 2016*, IEEE, 2016, pp. 1–5.
- [22] Sezgin, M., and Sankur, B., “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic imaging*, Vol. 13, No. 1, 2004, pp. 146–166.
- [23] Alimo, S. R., Beyhaghi, P., and Bewley, T. R., “Optimization combining derivative-free global exploration with derivative-based local refinement,” *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, IEEE, 2017, pp. 2531–2538.
- [24] Beyhaghi, P., and Bewley, T., “Implementation of Cartesian grids to accelerate Delaunay-based derivative-free optimization,” *Journal of Global Optimization*, Vol. 69, No. 4, 2017, pp. 927–949.